The Claims

1. (Previously presented) A computer-implemented method, comprising:

creating a software development kit object (SDK object) for at least some of a plurality of development files in a source operating system that includes development files and components;

identifying features of the source operating system to be included in a modularized system that is a subset of the source operating system;

tracing dependencies in a dependency model correlating to the source operating system that uses the SDK objects to identify SDK objects corresponding to development files that are required to support the identified features;

selecting the development files that correspond to the identified SDK objects; and

exporting the selected development files to a software development kit (SDK) that supports development of applications for use with the modularized system.

2. (Previously presented) The method as recited in claim 1, wherein the modularized system is a modularized system that includes a subset of development files and components of the source operating system.

3. (Previously presented) The method as recited in claim 1, further comprising generating the dependency model using the SDK objects.

4. (Previously presented) The method as recited in claim 1, further comprising generating the dependency model by:

identifying dependencies between development files;

creating SDK objects for the development files; and

for all the development files, if a first development file depends on a second development file, including a reference in a first SDK object associated with the first development file to a second SDK object associated with the second development file.

5. (Previously presented) The method as recited in claim 1, wherein the development files further comprise at least one or more of the following types of files: library files, documentation files, header files.

6. (Previously presented) The method as recited in claim 1, wherein the tracing dependencies further comprises tracing references from a first SDK object associated with a feature to at least a second SDK object and, if the second SDK object includes a reference to a third SDK object, tracing the reference to the third SDK object.

7. (Previously presented) The method as recited in claim 1, wherein the tracing dependencies further comprises:

identifying a data object associated with each identified feature, the data object being associated with a component of the source operating system; and

if a data object includes a reference to a first SDK object, tracing the reference to the first SDK object and tracing references, if any, from the first SDK object to a second SDK object.

8. (Previously presented) The method as recited in claim 1, wherein creating the SDK objects further comprises:

naming a data object having a type that identifies the data object as being an SDK object;

including at least one reference in a first SDK object, the reference pointing to a second SDK object that is required by the first SDK object to function properly; and

repeating the previous steps for each development file to be exposed in the SDK.

9. (Original) The method as recited in claim 8, further comprising including an export list in the first data object that identifies one or more functions that may be exposed by the development file associated with the SDK object.

10. (Previously presented) The method as recited in claim 1, wherein the tracing dependencies in a dependency model further comprises tracing dependencies beginning with a first set of SDK objects that are associated with the features to subsequent sets of SDK objects on which the first set of SDK objects depend.

11. (Previously presented) The method as recited in claim 1, wherein the tracing dependencies in a dependency model further comprises:

tracing dependencies beginning with data objects that are associated with the features, the data objects having references to one or more SDK objects in a first set of SDK objects, the first set of SDK objects having one or more reference to one or more SDK objects in a second set of SDK objects; and

wherein the first set of SDK objects depend on the second set of SDK objects for the first set of SDK objects to function.

12. (Previously presented) The method as recited in claim 1, wherein the exporting further comprises storing the selected development files on one or more computer-readable media.

13. (Currently amended) One or more computer-readable media containing computer-executable instructions that, when executed on a computer, perform the following steps:

selecting one or more developmental files associated with a source operating system that are required to support features selected for a modularized system that is a subset of the source operating system by tracing a dependency model that includes SDK objects associated with the one or more development files; and

exporting the selected development files to a software development kit (SDK).

14. (Original) The one or more computer-readable media as recited in claim 13, wherein the selecting one or more development files further comprises:

selecting one or more features of the source operating system to be included in the modularized system, each of the one or more features having a data object associated therewith;

tracing dependencies from the data objects to software development kit (SDK) objects that are associated with development files in order to identify SDK objects that are associated with development files to be included in a software development kit associated with the modularized system; and

selecting a development file that is associated with an identified SDK object.

15. (Previously presented) The one or more computer-readable media as recited in claim 13, wherein the selecting one or more development files further comprises:

selecting one or more features of the source operating system to be included in the modularized system, each of the one or more features having a software development kit object (SDK object) associated therewith;

tracing dependencies from the SDK objects associated with the features to SDK objects that are associated with development files in order to identify SDK objects that are associated with development files to be included in a software development kit associated with the modularized system; and

selecting a development file that is associated with an identified SDK object.

16. (Original) The one or more computer-readable media as recited in claim 13, further comprising creating the dependency model by creating the SDK objects to model dependencies between the development files associated with the SDK objects.

17. (Original) The one or more computer-readable media as recited in claim 16, wherein the creating the SDK objects further comprises:

for each development file in the source operating system that may be included in the modularized system, identifying a data object as being an SDK object that corresponds to a development file;

if a first development file associated with an SDK object requires the presence of a second development file to function properly, including a reference in a first SDK object associated with the first development file to a second SDK object.

18. (Original) The one or more computer-readable media as recited in claim 13, wherein the development files are of one or more of the following types of files: header files; library files; and/or documentation files.

19. (Previously presented) A method, implemented in a system, the method comprising:

identifying features in a source operating system to be included in a modularized system;

selecting development files to be included in an SDK by tracing dependencies in a dependency model beginning with data objects associated with the identified features; and

exporting the selected development files; and

wherein the development files are files required to support development of applications to work with the modularized system.

20. (Original) The method as recited in claim 19, further comprising generating the dependency model using the SDK objects.

21. (Original) The method as recited in claim 19, further comprising:

identifying dependencies between development files;

creating SDK objects for at least some of the development files;

for all development files that have an SDK object associated with it, if a first development file depends on a second development file, including a reference in a first SDK object associated with the first development file to a second SDK object associated with the second development file; and

wherein the identifying and creating generates the dependency model, and the selecting development files further comprises tracing references in SDK objects to determine the development files to select.

22. (Original) The method as recited in claim 19, wherein the development files include one or more of the following types of files: header files; library files; and/or documentation files.

23. (Original) The method as recited in claim 19, wherein the exporting further comprising outputting the selected development files to one or more computer-readable medium.

24. (Original) The method as recited in claim 19, further comprising generating the dependency model utilizing the SDK objects.

25. (Currently amended) One or more computer-readable storage media containing computer-executable instructions that, when executed on a computer, perform the following steps:

selecting one or more software development kit objects (SDK object), each SDK object being associated with a developmental file in a source operating system that is required to support one or more features of the source operating system selected for inclusion in a modularized system that is a subset of the source operating system; and

filtering a master SDK header file to determine developmental files to include with a software development kit (SDK) that is used with the modularized system to allow software to be developed to work with the modularized system.

26. (Original) The one or more computer-readable media as recited in claim 25, wherein the filtering further comprises:

for each selected SDK object, searching for a label in the master SDK header file that is the same name as the SDK object;

if the label is found, enabling a section of code associated with the label; and

wherein the enabling the section of code associated with the label enables the appropriate development file associated with the SDK object to be included in the SDK.

27. (Original) The one or more computer-readable media as recited in claim 25, wherein the filtering further comprises:

executing the master SDK header file that includes labels that are the same names as the SDK objects, each of the labels being associated with a section of code;

when encountering a label, determining of an SDK object of the same name as the label has been selected for inclusion in the SDK;

if the SDK object has been selected, enabling the section of code associated with the label;

wherein the section of code associated with the label enables the appropriate development file associated with the SDK object to be included in the SDK.

28. (Original) The one or more computer-readable media as recited in claim 25, wherein the development files further comprise one or more of the following types of files: library files; header files; and/or documentation files.

29. (Original) The one or more computer-readable media as recited in claim 25, wherein the selecting one or more SDK objects further comprises:

tracing SDK object dependencies using a dependency model, the tracing being from one or more data objects associated with the selected features to one or more SDK objects; and

selecting each SDK object encountered in the tracing process.

30. (Original) The one or more computer-readable media as recited in claim 29, wherein the data objects associated with the features are SDK objects.

31. (Original) The one or more computer-readable media as recited in claim 29, wherein the data objects associated with the features also reference data objects associated with one or more components of the source operating system.

32. (Original) The one or more computer-readable media as recited in claim 25, further comprising exporting the developmental files derived from filtering the master SDK header file.

33. (Original) The one or more computer-readable media as recited in claim 32, wherein the exporting further comprises storing the developmental files on one or more computer-readable media.

34. (Original) A system, comprising:

an SDK object generator configured to generate one or more SDK objects, each SDK object being associated with a development file of a source operating system;

a feature identification module that can be used to select desired features from the source operating system to be included in a modularized system that is a subset of the source operating system;

a dependency tracer configured to trace references from data objects associated with the selected features to identify SDK objects associated with the development files required to support the selected features; and

an export module configured to export the development files associated with the identified SDK objects.

35. (Original) The system as recited in claim 34, wherein the SDK object generator is further configured to allow generation of a first SDK object that includes at least one reference to a second SDK object if a first development file associated with the first SDK object depends on the availability of a second development file associated with the second SDK object.

36. (Original) The system as recited in claim 35, wherein the SDK generator is further configured to allow generation of an SDK object that includes an export list that contains the names of functions of the source operating system, if any, that are exposed by the SDK object.

37. (Original) The system as recited in claim 34, wherein the data objects associated with the selected features are data objects that also reference components of the source operating system.

38. (Original) The system as recited in claim 34, wherein the data objects associated with the selected features are SDK objects.

39. (Original) The system as recited in claim 34, wherein the development files further comprise one or more files of the following types: header files; documentation files; and/or library files.

40. (Original) The system as recited in claim 34, wherein the export module is further configured to transmit the selected development files to one or more computer-readable media where the development files are stored.

41. (Original) The system as recited in claim 40, wherein the development files are stored in a software development kit.